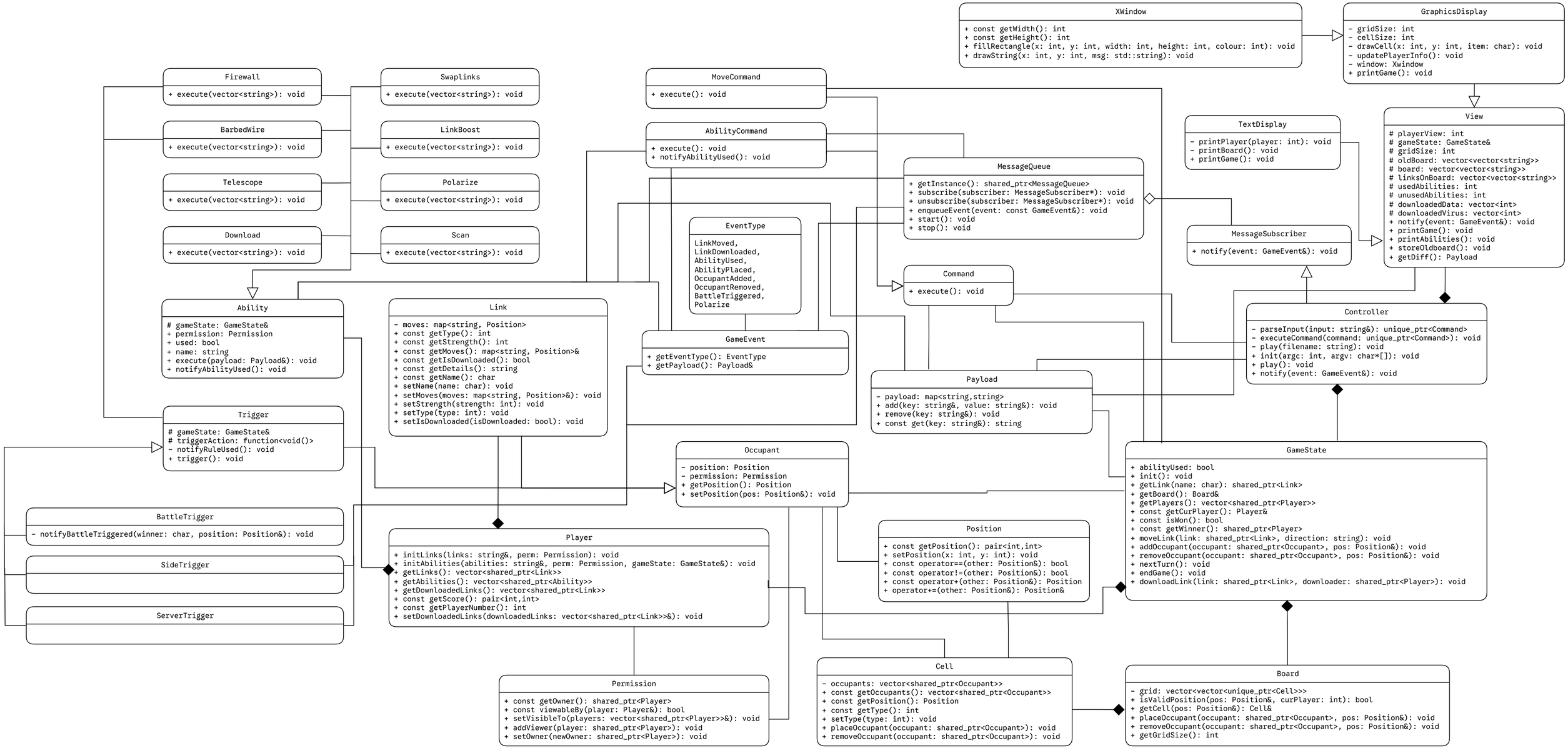
XWindow
+ const getWidth(): int
+ const getHeight(): int
+ fillRectangle(x: int, y: int, width: int, height: int, colour: int): void
+ drawString(x: int, y: int, msg: std::string): void
GraphicsDisplay
- gridSize: int
- cellSize: int
- drawCell(x: int, y: int, item: char): void
- updatePlayerInfo(): void
- window: Xwindow
+ printGame(): void
Firewall
+ execute(vector<string>): void
Swaplinks
+ execute(vector<string>): void
MoveCommand
+ execute(): void
BarbedWire
+ execute(vector<string>): void
LinkBoost
+ execute(vector<string>): void
AbilityCommand
+ execute(): void
+ notifyAbilityUsed(): void
TextDisplay
- printPlayer(player: int): void
- printBoard(): void
+ printGame(): void
View
playerView: int
gameState: GameState&
gridSize: int
oldBoard: vector<vector<string>>
board: vector<vector<string>>
linksOnBoard: vector<vector<string>>
usedAbilities: int
unusedAbilities: int
downloadedData: vector<int>
downloadedVirus: vector<int>
+ notify(event: GameEvent&): void
+ printGame(): void
+ printAbilities(): void
+ storeOldboard(): void
+ getDiff(): Payload
Telescope
+ execute(vector<string>): void
Polarize
+ execute(vector<string>): void
MessageQueue
+ getInstance(): shared_ptr<MessageQueue>
+ subscribe(subscriber: MessageSubscriber*): void
+ unsubscribe(subscriber: MessageSubscriber*): void
+ enqueueEvent(event: const GameEvent&): void
+ start(): void
+ stop(): void
Download
+ execute(vector<string>): void
Scan
+ execute(vector<string>): void
EventType
LinkMoved,
LinkDownloaded,
AbilityUsed,
AbilityPlaced,
OccupantAdded,
OccupantRemoved,
BattleTriggered,
Polarize
MessageSubscriber
+ notify(event: GameEvent&): void
Command
+ execute(): void
Ability
gameState: GameState&
+ permission: Permission
+ used: bool
+ name: string
+ execute(payload: Payload&): void
+ notifyAbilityUsed(): void
Link
- moves: map<string, Position>
+ const getType(): int
+ const getStrength(): int
+ const getMoves(): map<string, Position>&
+ const getIsDownloaded(): bool
+ const getDetails(): string
+ const getName(): char
+ setName(name: char): void
+ setMoves(moves: map<string, Position>&): void
+ setStrength(strength: int): void
+ setType(type: int): void
+ setIsDownloaded(isDownloaded: bool): void
GameEvent
+ getEventType(): EventType
+ getPayload(): Payload&
Controller
- parseInput(input: string&): unique_ptr<Command>
- executeCommand(command: unique_ptr<Command>): void
- play(filename: string): void
+ init(argc: int, argv: char*[]): void
+ play(): void
+ notify(event: GameEvent&): void
Payload
- payload: map<string,string>
+ add(key: string&, value: string&): void
+ remove(key: string&): void
+ const get(key: string&): string
Trigger
gameState: GameState&
triggerAction: function<void()>
- notifyRuleUsed(): void
+ trigger(): void
Occupant
- position: Position
- permission: Permission
+ getPosition(): Position
+ setPosition(pos: Position&): void
GameState
+ abilityUsed: bool
+ init(): void
+ getLink(name: char): shared_ptr<Link>
+ getBoard(): Board&
+ getPlayers(): vector<shared_ptr<Player>>
+ const getCurPlayer(): Player&
+ const isWon(): bool
+ const getWinner(): shared_ptr<Player>
+ moveLink(link: shared_ptr<Link>, direction: string): void
+ addOccupant(occupant: shared_ptr<Occupant>, pos: Position&): void
+ removeOccupant(occupant: shared_ptr<Occupant>, pos: Position&): void
+ nextTurn(): void
+ endGame(): void
+ downloadLink(link: shared_ptr<Link>, downloader: shared_ptr<Player>): void
BattleTrigger
- notifyBattleTriggered(winner: char, position: Position&): void
Position
+ const getPosition(): pair<int,int>
+ setPosition(x: int, y: int): void
+ const operator==(other: Position&): bool
+ const operator!=(other: Position&): bool
+ const operator+(other: Position&): Position
+ operator+=(other: Position&): Position&
SideTrigger
Player
+ initLinks(links: string&, perm: Permission): void
+ initAbilities(abilities: string&, perm: Permission, gameState: GameState&): void
+ getLinks(): vector<shared_ptr<Link>>
+ getAbilities(): vector<shared_ptr<Ability>>
+ getDownloadedLinks(): vector<shared_ptr<Link>>
+ const getScore(): pair<int,int>
+ const getPlayerNumber(): int
+ setDownloadedLinks(downloadedLinks: vector<shared_ptr<Link>>&): void
ServerTrigger
Permission
+ const getOwner(): shared_ptr<Player>
+ const viewableBy(player: Player&): bool
+ setVisibleTo(players: vector<shared_ptr<Player>>&): void
+ addViewer(player: shared_ptr<Player>): void
+ setOwner(newOwner: shared_ptr<Player>): void
Cell
- occupants: vector<shared_ptr<Occupant>>
+ const getOccupants(): vector<shared_ptr<Occupant>>
+ const getPosition(): Position
+ const getType(): int
+ setType(type: int): void
+ placeOccupant(occupant: shared_ptr<Occupant>): void
+ removeOccupant(occupant: shared_ptr<Occupant>): void
Board
- grid: vector<vector<unique_ptr<Cell>>>
+ isValidPosition(pos: Position&, curPlayer: int): bool
+ getCell(pos: Position&): Cell&
+ placeOccupant(occupant: shared_ptr<Occupant>, pos: Position&): void
+ removeOccupant(occupant: shared_ptr<Occupant>, pos: Position&): void
+ getGridSize(): int